




Clean Software.  
Guaranteed.

Disciplined Test Process Drives SDLC  
By Sujatha C

## What is planned?



Clean Software.  
Guaranteed.

- Part-1 (practical experience)
  - Impact of documentation not in synch/ vague requirements
  - Effective change management
  - Role of SCM
- Part-2 (good practices recommended)
  - Where we normally face problem?
  - Improve incoming quality of build to QA
  - Disciplined approach to defect removal - Levels of testing
  - Test process ad-hoc to disciplined

## Part-1




Clean Software.  
Guaranteed.

### Three main issues

- Requirements not captured properly
- Change requests not documented
  - Impact analysis of CR's do not happen
- Release management not in place
  - Version control of work products not happening

## Testing & process go hand-in-hand



Clean Software.  
Guaranteed.

- Testing with out proper requirements will not uncover all defects
- Testing without change request impact analysis will lead to defect escapes
- Testing completed but releasing the wrong product version to the customer will lead to severe problems



## Process can be customized

- Requirements can be captured as simple bullets or in an excel sheet
- Requirements need to be reviewed with customer and internally with the testing team
- Attempt to freeze requirements before proceeding with other steps in the SDLC (75-80%)



## Process to be monitored

- Form a SEPG group
- Group to consist of Sr. and Jr. engineers
- Goal for this group
  - To implement the process
  - To ensure it is followed rigorously
  - Conduct planned and surprise audits

## Process can be evolved



- All users must feel comfortable with the process steps
- Adherence will become natural
- All process users (employees) input/feedback is a must to freeze the process steps
- A process handbook with templates and samples can be formulated

## Training



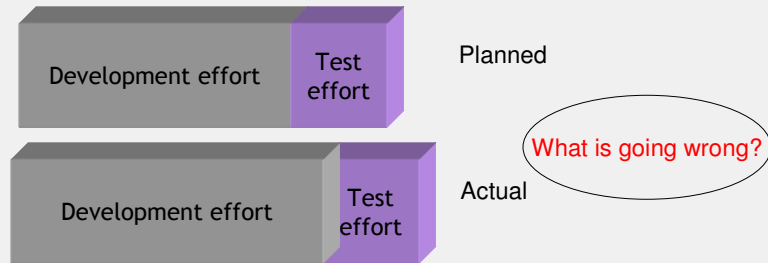
- Educate the users about following the process thru out the SDLC
- Importance of levels of testing and about defects at different levels
- Share the product quality success with the users to motivate them.

## Part-2

Clean Software.  
Guaranteed.

Where we normally face problem?

A cycle of test involves development of set of requirements and tested as planned

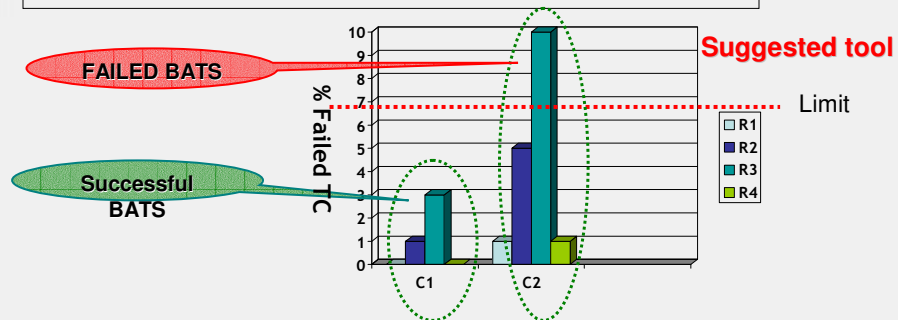


## Improved incoming quality

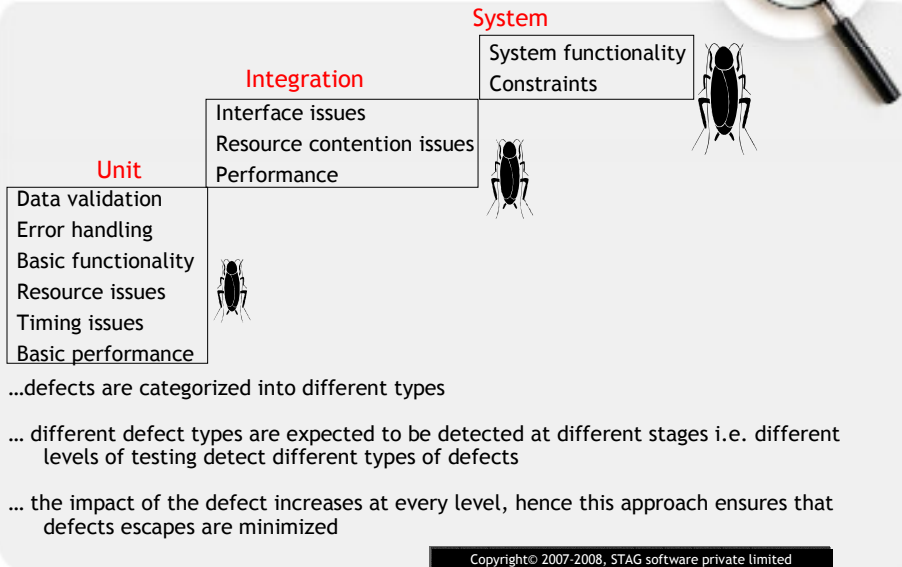
Clean Software.  
Guaranteed.

### Control incoming quality to test cycle

- Implement Basic Acceptance Test Scripts
- Define scope for BATS involving development team
- Set a goal to control BATS failure
- Introduce root cause analysis process



## A disciplined approach to defect removal

Clean Software.  
Guaranteed.

## Test process - Ad-hoc to disciplined

Clean Software.  
Guaranteed.

## Check yourself

- Do you know how many system requirements you need to test?
- Do you know what non-functional quality factors we need to test?
- Do you know how much time it will take to test your product?
- Do you know how much to re-test because of changes done to code?
- Do you know customers priority on system requirements?

## Does this help?

### Suggested template

Req #	Short description	Priority	# TS	#TC	Interface with req.

- Add required columns for tracking project status**
- Sort as per customer priority**
- All reports with respect to customer priority**
- Estimation accuracy improves**

## Case study

**Company:** A software Product Company developing ERP solution to meet the demands of small and medium scale Pharmaceutical, Chemical and Food processing industries

**Product:** ERP solution complies with **Current Good Manufacturing Practices (cGMP)** and requirements of **International Regulatory Bodies** such as US FDA, EDQM, TGA, MHRA, MCC, etc.

The major components of the system are

- Human resource and payroll management
- Financial Accounting and Management
- Material Management
- Production planning
- Quality Management
- Plant Maintenance
- Sales and Distribution

# STAG contribution

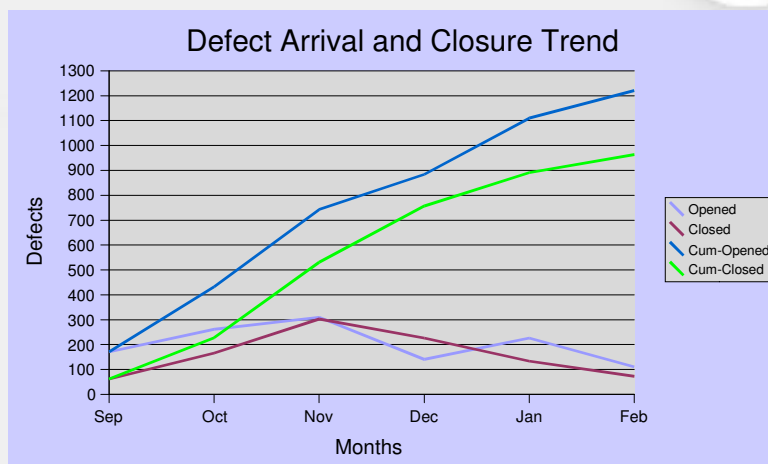


## What we implemented

Understood requirements by interacting with functional experts

- ✓ By using our techniques
  - ✓ Designed module level test cases and executed
  - ✓ Designed module integration level test cases and executed
  - ✓ Designed system level end-to-end business scenarios for regression testing
- ✓ Participated in multiple cycles of testing in phased delivery model
- ✓ Imparted training on how to capture requirements, about importance of Unit level testing, etc.
- ✓ Introduced SDLC and testing process
- ✓ Significantly helped and participated in organization SDLC process improvement activities
- ✓ Played a major role being part of SEPG (Software Engineering Process Group) team and thus ensured that Process and testing produced a quality product

# First release after process implementation!



# Tangible benefits



- Using well documented test cases estimation for testing effort improved
- Schedule variance brought under control to 5%
- Unit level defect escapes to system level reduced drastically
- Defect escapes to the customer came down drastically
- Product quality results shared with developers and testers helped boost their confidence
- Process followers were convinced that disciplined process and testing will lead to quality product every time!

# Thank You



Questions?

Thank You